Constructive contextual modal judgments for reasoning from open assumptions

Giuseppe Primiero*

Centre for Logic and Philosophy of Science
University of Ghent
Blandijnberg 2, 9000 Gent (Belgium)
Giuseppe.Primiero@UGent.be

Abstract. Dependent type theories using a structural notion of context are largely explored in their applications to programming languages, but less investigated for knowledge representation purposes. In particular, types with modalities are already used for distributed and staged computation. This paper introduces a type system extended with judgmental modalities internalizing epistemically different modes of correctness to explore a calculus of provability from refutable assumptions.

1 Introduction

Constructive logics use proofs as first-class citizens to define the notion of truth. Dependent truth is easily interpreted in a contextual reading of provability, as in Martin-Löf Type Theory.[1] In such a system one distinguishes between proposition A and judgment A *true*, justified by an appropriate proof term $a : A$. Correspondingly, contextual truth allows formulae of the form $\Gamma \vdash a : A$, with $\Gamma = [x_1 : A_1, \ldots, x_n : A_n]$ and a a proof of A under appropriate substitutions $[x_1/a_1 : A_1, \ldots, x_n/a_n : A_n] \vdash a : A$. Hypothetical truth is thus reduced to dependent closed constructions, hypotheses are obtained by abstracting on the relevant proofs and ultimately grounded on the primitive notion of premise (known judgment). Computationally this corresponds to β-reduction for proof terms and the evaluation of codes in a program. The modal formulation of contextual calculi is the next obvious step. Along with the standard intuitionistic translation of K and the constructive version of $S4$,[2] a weaker format to accommodate the notion of context is given by the possible-world semantics CK in [13], sound and complete with respect to the natural deduction interpretation of [5].[3] Re-

* Post-Doctoral Fellow of the Research Foundation - Flanders. Associate Researcher IEG - Oxford University. The author wishes to thank the participants at the Logic Seminar, Helsinki University and at the Hypo Project, IHPST, Paris for comments; in particular Sara Negri and Peter Schröder-Heister for their useful insights; and three anonymous referees for helpful comments.

[1] See [11], [12].

[2] See for example [24], [2], [1].

[3] This system is the most basic modal logic of contexts, with formulas $\mathtt{ist}(k, A)$ that read "A is true in context k" and it satisfies a multi-modal K fragment of a Propositional Logic of Contexts.

cently, contextual modal type theories for programming languages and further research in linguistics and hardware verification have been formulated, especially to model staged and distributed computation.[4]

In the present paper, a modal type system is used to formalize epistemic processes under refutable assumptions. Our starting point is the constructive reading of the notion of truth as existence of a verification, to design a type-theorethical format for the epistemic notion of *verification under open assumptions*. The notion of truth up to refutation recalls a sensible topic for constructive logics, based on the meaning of intuitionistic negation.[5] The present paper dwells on the foundational idea that truth is admissible up to a counter-example.[6] Admissible truths are literally satisfied by the logical concept of assumption, intended as a computational term which is not presented together with an appropriate β-redex. The related constructive modal type system variates on a theme first proposed in [18] and later expanded in [16]. In section 2 of the paper I provide a variant interpretation of the basic system of constructive type-theory that links hypotheses and refutable contents; in section 3 a modal type system is designed that preserves refutability. Conclusive remarks set the next steps of this research.

2 A system for proven and refutable contents

Describing realistic knowledge processes requires explaining hypotheses as contents whose truth is declared, but whose refutation is not ruled out.[7] The corresponding logical notion is that of an (open) assumption which needs to be justified independently from proven contents to be integrated in the constructive definition of truth.

To obtain this, we start from a polymorphic language containing one basic sort *type* for categorical constructive judgments with corresponding term constructors a, b; and one sort $type_{inf}$ (*information type*) for judgments in a context of refutable conditions, with corresponding variable constructors x_1, x_2. Judgements of the first sort induce a constructive notion of truth (*true*), the second ones a weaker predicate of truth up to verification ($true^*$). Identity of terms hold within *type*, and its constructors are composed standardly by way of listing, application, abstraction and pairing to define connectives and quantifiers: $\wedge, \vee, \rightarrow, \forall, \exists$. In particular: $\rightarrow$ is material implication obtained by application of two already obtained constructions, namely by an application $a(b)$ of the construction a of the antecedent to the construction b of the consequent, which can

[4] See e.g. [4], [14], [15].

[5] For the standard intuitionistic meaning explanation of negation, indirect proofs as *reductio ad absurdum* are standardly not admitted, whereas the usual intuitionistic absurdity rule interprets the classical *ex falso quodlibet*. See e.g. [23, p. 40].

[6] This remembers the notion of 'pseudo-truth' introduced in [10] for double-negated classical formulae reducible to intuitionistic ones.

[7] Formally, it expresses processual information without appropriate computational instructions, the same intuition behind the explanation of partial evaluation, where a function program considers part of its input code as given. Cf. [9].

be seen as a λ-term presented *together with* one of its α-terms.[8] $\forall$ abstracts from enumerable sets of equivalent constructions; $\exists$ is justified by paired constructions. Admissibility of the $type_{inf}$ sort is defined in two steps: first, the construction $a\!:\!A$ establishes the inadmissibility of $\neg A$ in further contexts; secondly, a missing construction for $(A \rightarrow \bot)$ allows assumption formation $x : A$. Variables are unique in context for $type_{inf}$; abstraction and conversion on variables define respectively function formation and reduction to *type*. Material implication is therefore distinct from function formation $\supset$, the latter being given by abstraction on the admissible construction for the antecedent. Proof terms can occur both in and outside of a context, proof variables hold only in contexts. Types are typically propositions and judgments are built by declaration of *true* or *true**. The syntactically introduced semantic notions extend the standard format of Constructive Type Theory. In the following, we omit for brevity the identity rules that define Reflexivity, Symmetry and Transitivity on types; the explanation of modal contexts is left to the next section. Our syntax is as follows:

Types := $A\ type; A\ type_{inf}$;
Propositions := $A; A \wedge B; A \vee B; A \rightarrow B; (\exists a_i\!:\!A_i)B; (\forall a_i\!:\!A_i)B; A \supset B$;
Proof terms := $a : A; (a, b); a(b); \lambda(a(b)); <a, b>$;
Proof variables := $x\!:\!A; (x(b)); (x(b))(a)$;
Contexts := $\Gamma, x\!:\!A; \Gamma, a\!:\!A; \Box\Gamma; \Diamond\Gamma$;
Judgments := $A\ true; A\ true^*; \Gamma \vdash A\ true; \Diamond(A\ true); \Box(A\ true)$.

Introduction Rules express the validity of semantic judgements ($true/true^*$) from appropriate syntactic constructions ($a/x\!:\!A$); quantifiers introduction and conversion determine the validity of new typing judgement (*A type*) from previously defined ones.

$$\frac{a\!:\!A}{A\ type}\ \text{Type Formation} \qquad \frac{a\!:\!A}{A\ true}\ \text{Truth Definition}$$

$$\frac{a\!:\!A}{l(a)\!:\!A \vee B\ true}\ Left I\vee \qquad \frac{b\!:\!B}{r(b)\!:\!A \vee B\ true}\ Right I\vee$$

$$\frac{a\!:\!A \quad b\!:\!B}{(a, b)\!:\!A \wedge B\ true}\ I\wedge \qquad \frac{a\!:\!A \quad A\ true \vdash b\!:\!B}{a(b)\!:\!A \rightarrow B\ true}\ I \rightarrow (Implication)$$

$$\frac{a_1\!:\!A_i, \ldots, a_n\!:\!A_i \quad [A_i\ true] \vdash b\!:\!B \quad \lambda((a_i(b))A, B)}{(\forall a_i\!:\!A_i)B\ type}\ I\forall$$

$$\frac{a_1\!:\!A_i, \ldots, a_n\!:\!A_i \quad [a_i\!:\!A_i] \vdash b\!:\!B \quad [a_i\!:\!A_i] \vdash b\!:\!B}{(\exists a_i\!:\!A_i)B\ type}\ I\exists$$

[8] A literal interpretation of intuitionistic implication á la Heyting. This implements in the language the standard (meta-)reasoning requiring the substitution of the construction of the antecedent to be performed in order the construction of the consequent to be obtained and it recalls ideas mentioned by Martin-Löf and the calculus of types with explicit substitutions presented in [22].

$$\frac{a:A}{\neg A \to \bot}\, I\bot$$

An elimination rule on the $\bot$-rule would validate a double-negation elimination, which is avoided by a non-standard extension to functional expressions. Formally, a dependent judgment is nothing else than a functional relation among expressions: if A *type* holds, then a construction of a new type B is possible by considering the latter as a family of sets over some $x:A$ such that $x:A \vdash B$ *type* whenever the substitution $[x/a]$ is performed.[9] A new task is to admit no explicit evaluation on such formulae, extending the system with the new type format $type_{inf}$. Formulas of the information type are introduced by proof variables; a judgment $A\ type_{inf}$ is justified by running a test on previous derivations such that it checks no construction for $\neg A$ *type* to be given:

$$\frac{\neg(A \to \bot)}{A\ type_{inf}}\quad \text{Informational Type Formation}$$

$$\frac{A\ type_{inf} \quad x:A}{A\ true^{*}}\quad \text{Hypothetical Truth Definition}$$

The judgment $\neg(A \to \bot)$ says that there exists no construction for $A\ true \to \bot$. Its combination with $type_{inf}$ formation does not imply that from $\neg(A \to \bot)$ follows $a : A$: the latter justification is kept entirely constructive and therefore cannot be given by indirect proof. The second rule says that provided A can be admitted as a $type_{inf}$, a weak truth-predicate $true^{*}$ (true up to refutation) is inferred by assuming a construction for A exists: it can be seen as a placeholder for ungrounded truth.[10] On this interpretation one defines functional expressions:

[9] The type checking will require first well-formedness of A, secondly evaluation to a current environment for extraction of variable terms, thirdly construction for the variable in that environment, and finally evaluation of the variable and the formulation of the binding expression to a value for that environment. The generalization to multiple dependence being allowed, terms for $[x_1:A_1,\ldots,x_n:A_n] \vdash B$ *type* are evaluated to normal forms (eventually: weak head normal forms, explicit substitutions, closures) in order the predication B *type* to be valid.

[10] Our admissibility rule for the $type_{inf}$ sort interprets the distinction between intensionality and extensionality of types as treated e.g. in [17]: expressions are treated intensionally being subject only to α-conversion; terms are treated extensionally, being additionally subject to β and η-conversion. Similarieties can also be found with the Lax modality defined in a propositional intuitionistic logic in [6]: the modal formula $\circ\phi$ expresses the inhabitation of ϕ in the context of a number of assumptions holding in a stronger theory. The theory designs two distinct and dual contexts: one where the formula is true only in certain worlds where appropriate constraints hold, the other only where constraints are false. The former is the partial element lifting and the latter the exception lifting for the type formula ϕ at hand. See [6, p.65]. Our double-negated typing might be seen as a way of admitting the first kind of constraints, up to proving that the second kind holds.

$$\frac{A\ type_{inf} \quad x\!:\!A \vdash B\ type_{inf}}{x\!:\!A \vdash B\ true^{*}}$$

which says that B is true up to a refutation of A. The weak truth predicate induces the standard dependent functional construction by abstraction; β-conversion provides the appropriate translation to standard dependent type formation by application:

$$\frac{A\ type_{inf} \quad x\!:\!A \vdash B\ true^{*}}{((x)b) : A \supset B\ true}\ Functional\ Abstraction$$

$$\frac{A\ type_{inf} \quad x\!:\!A \vdash B\ type_{inf} \quad a\!:\!A}{(x(b))(a) = b[a/x]\!:\!B\ type[a/x]}\ \beta - conversion$$

3 Contextual Modal Type Theory for verification and refutation

The different notions of truth are internalized in our system by the use of epistemic modalities. Previous modal versions of type theory ([18, 16]) use propositional modalities to speak about dependent truth via additional judgments: "proposition 'A is necessary' is true" ($\Box A\ true$) and "proposition 'A is possible' is true" ($\Diamond A\ true$). Such system satisfies the condition of reduction of hypotheses to closed constructions by distinguishing assumptions of truth and valid assumptions. If ($\Box A\ true$) means the truth of A in a world in which we know nothing (validity), ($\Diamond A\ true$) means that there is nothing else we can say about the world in which this happens (so that nothing else can be assumed). This system satisfies a $S4$ normal modal logic.

In the present system, modalities are judgmental operators:[11] $\Box(A\ true)$ says that A is true and has no refutable conditions (either there are none, or all of them have been secured); if $A\ true$ holds, it also holds under refutable data being added, by definition no declaration $\neg A\ type_{inf}$ being allowed if $a\!:\!A$ is formulated. This makes A verified in *any* extension of the empty context. $\Diamond(A\ true)$ says that A is true in those epistemic states where conditions are not refuted. Our type-theoretical language defines a derivability relation that can be simulated in a model-theoretical setting with a verification function over ordered models.[12] Such models are provably equivalent to those of a contextual version of KT with $\Box$ and $\Diamond$, hence inducing an equivalence with the fragment of constructive $S4$ with possibility and without iterations.[13]

---

[11] For more on the philosophical justification of this notion of judgmental modalities, see [21].

[12] See [20].

[13] See [1]. The models corresponding to *type* are those with the categorical verification function and the satisfied dependent ones: these are both reflexive and transitive over the preorder and satisfy axiom $T_{\Box} : \Box A \rightarrow A$. The models corresponding to the $type_{inf}$ sort are models with refutable assumptions; they deal with the possibility operator by means of the axiom $T_{\Diamond} : A \rightarrow \Diamond A$ and the $\Diamond$-introduction rule: $\Gamma, \Gamma' \vdash A \Rightarrow \Box\Gamma, \Diamond\Gamma' \vdash \Diamond A$.

A premise and a hypothesis rule introduce the truth predicates (both rules can have $\Gamma, \Delta = \{\emptyset\}$):

$$\frac{}{\Gamma, a\!:\!A, \Delta \vdash A\ true}\ \text{Premise Rule} \qquad \frac{}{\Gamma, x\!:\!A, \Delta \vdash A\ true^*}\ \text{Hypothesis Rule}$$

Definition 1 (Definition of (Local) Validity).

1. *If A true then A is valid.*
2. *If A is valid then $\Gamma \vdash A$ true, for every Γ.*
3. *If A true* then A is locally valid in view of some $\Gamma \vdash A$ true.*

Modalities are internalized by appropriate formation rules from categorical and hypothetical judgments:

$$\frac{a\!:\!A}{\Box(A\ true)}\ \Box\text{-Formation} \qquad \frac{x\!:\!A}{\Diamond(A\ true)}\ \Diamond\text{-Formation}$$

The inference to truth of contextual judgements requires verification of assumptions. To this aim, modalities are now generalized to contextual formulas.[14]

Definition 2 (Necessitation Context). *For any context Γ, $\Box\Gamma$ is given by $\bigcup\{\Box A\ true \mid$ for all $A \in \Gamma\}$.*

Definition 3 (Normal Context). *For any context Γ, $\Diamond\Gamma$ is given by $\bigcup\{\circ A\ true \mid \circ = \{\Box, \Diamond\}$ and $\Diamond A$ true for at least one $A \in \Gamma\}$.*

The introduction of judgmental $\Box$ is allowed under verification of judgments in context, its elimination rule induces a valid proposition:[15]

$$\frac{\Gamma \vdash A\ true}{\Box\Gamma \vdash \Box(A\ true)}\ I\Box \qquad \frac{\Box\Gamma \vdash \Box(A\ true) \qquad \Delta, a\!:\!A \vdash b\!:\!B}{\Gamma, \Delta \vdash B\ true}\ E\Box$$

where $\Box\Gamma$ iff $[x_i/a_i] : A_i, \forall A_i \in \Gamma$, as by Definition 2. Local validity is in turn defined by introduction and elimation rules for the $\Diamond$-operator:

$$\frac{\Gamma, x\!:\!A \vdash B\ true^*}{\Box\Gamma, \Diamond(A\ true) \vdash \Diamond(B\ true)}\ I\Diamond \qquad \frac{\Gamma, \Delta \vdash A\ true^* \quad \Box\Gamma, \Diamond(A\ true) \vdash \Diamond(B\ true)}{\Gamma, \Delta \vdash B\ true^*}\ E\Diamond$$

[14] In various literature in modal logic, *Necessitation* and *Normal Context* are usually called *Global* and *Local Context*. This distinction is crucial for derivability under assumption in modal languages, involving the validity of the Deduction Theorem, see [8]. I have strenghtened here the reasoning, by obtaining modal judgments (rather than formulae) from the preservation/verification of assumptions. Cf. [7].

[15] This is the crucial difference with the system introduced in [18], where $\Box A$ expresses validity but it can be introduced under hypotheses. In the comparison with the system in [5], the obvious similarity is that the therein contained modality $\Box_k$ satisfies the same principle of our $I\Box$, namely it builds-in the substituitions needed for formulas in contexts. On the other hand, the propositional format does not require any $\Diamond$ operator, its role being syntactically satisfied by standard contexts.

The introduction rule shows the dependency of possible contents from refutable conditions; the corresponding elimination uses this information to infer further possible knowledge under the condition expressed by Definition 3.

Substitution of variables by constants is as usual indicated by $[x/a]B$ as the substitution of occurrences of x in B by a; in our system this gives the relation between verification and truth and the modal version shows that term substitution satisfies the inclusion of $\Diamond$ in $\Box$:

Theorem 1 (Substitution on terms).

1. *If* $\Gamma, x:A, \Delta \vdash B\ true^*$ *and* $\Gamma, \Delta \vdash a:A$*, then* $\Gamma, \Delta \vdash [x/a]B\ true$.
2. *If* $\Box\Gamma, \Diamond(A\ true), \Box\Delta \vdash \Diamond(B\ true)$ *and* $\Box\Gamma, \Box\Delta \vdash \Box(A\ true)$*, then* $\Box\Gamma, \Box\Delta \vdash \Box(B\ true)$.

Proof. 1. by induction on the first given derivation, using the Hypothesis Rule and the inclusion of $B\ true^*$ in $B\ true$; from the second premise all occurrences of A are declared *type*, in particular those in $\Gamma, \Delta \vdash B\ true^*$ by β-conversion, then $B\ true$ follows as valid in any extension of Γ, Δ. 2. again by induction on the first given derivation: by $E\Diamond$ on the first premise one obtains an occurrence of $x:A$, using β-conversion on $A\ true^*$ one obtains $B\ true$ in the second premise; by $I\Box$ one finally obtains $\Box(B\ true)$. $\Box$

β-reduction and η-expansion, i.e. local inversion of modal rules hold; theorem 1 is crucial to this aim together with the structural properties of our system:

Theorem 2 (Weakening). *The inference systems satisfies Weakening:*

1. *If* $\Gamma \vdash B\ true$*, then* $\Gamma, a:A \vdash B\ true$.
2. *If* $\Gamma \vdash B\ true^*$*, then* $\Gamma, x:A \vdash B\ true^*$.
3. *If* $\Box\Gamma \vdash \Box(B\ true)$*, then* $\Box\Gamma, \Box(A\ true) \vdash \Box(B\ true)$.
4. *If* $\Diamond\Gamma \vdash \Diamond(B\ true)$*, then* $\Diamond\Gamma, \Diamond(A\ true) \vdash \Diamond(B\ true)$.

Proof. By induction on derivations: in 1. uses the Premise Rule; in 2. uses the Hypothesis Rule; in 3. uses $I\Box$, in 4. uses $I\Diamond$.

Theorem 3 (Contraction). *The inference system satisfies Contraction:*

1. *If* $\Gamma, a_1:A, a_2:A \vdash B\ true$*, then* $\Gamma, a:A \vdash [a_1 \approx a_2/a]B\ true$.
2. *If* $\Gamma, x_1:A, x_2:A \vdash B\ true^*$*, then* $\Gamma, x:A \vdash [x_1 \approx x_2/x]B\ true^*$.
3. *If* $\Box\Gamma, a_1:A, a_2:A \vdash \Box(B\ true)$*, then* $\Box\Gamma, \Box(A\ true) \vdash \Box(B\ true)$.
4. *If* $\Box\Gamma, x_1:A, x_2:A \vdash \Diamond(B\ true)$*, then* $\Box\Gamma, \Diamond(A\ true) \vdash \Diamond(B\ true)$.

Proof. By induction on derivations: Refleflexivity and Symmetry for proof terms in 1.; unicity of proof variables for $type_{inf}$ in 2.; in addition Truth Definition and $I\Box$ for 3.; Hypothetical Truth Definition and $I\Diamond$ for 4..

Theorem 4 (Exchange). *The inference system satisfies Exchange:*

1. *If* $\Gamma, a_1:A, a_2:A \vdash B\ true$*, then* $\Gamma, a_2:A, a_1:A \vdash B\ true$.

2. *If* $\Gamma, x_1 : A, x_2 : A \vdash B\ true^*$, *then* $\Gamma, x_2 : A, x_1 : A \vdash B\ true^*$.
3. *If* $\Box\Gamma, a_1 : A, a_2 : A \vdash \Box(B\ true)$, *then* $\Box\Gamma, a_2 : A, a_1 : A \vdash \Box(B\ true)$.
4. *If* $\Box\Gamma, x_1 : A, x_2 : A \vdash \Diamond(B\ true)$, *then* $\Box\Gamma, x_2 : A, x_1 : A \vdash \Diamond(B\ true)$.

Proof. By induction and using the same properties on terms and variables as for Contraction.

Local inversion of modal rules is finally shown. Soundness by local reduction on $\Box(A\ true)$:

$$\dfrac{\dfrac{\begin{matrix}D_1\\ \Gamma \vdash A\ true\end{matrix}}{\Box\Gamma \vdash \Box(A\ true)}\,I\Box \qquad \begin{matrix}E\\ \Delta, a : A \vdash b : B\end{matrix}}{\Gamma, \Delta \vdash B\ true}\,E\Box \quad \Rightarrow_{Redex} \quad \begin{matrix}D_2\\ \Gamma, \Delta \vdash B\ true\end{matrix}$$

D_2 is obtained from D_1 and E by Theorem 1: a proof term for A is induced from Γ in D_1, in turn providing a proof term for B in E. In computational terms, this rule formalizes β-reduction of B (value) with respect to all occurrences of its procedures (codes) in A. Completeness by local expansion on $\Box(A\ true)$:

$$\underline{\begin{matrix}D_1\\ \Box\Gamma \vdash \Box(A\ true)\end{matrix}} \quad \Rightarrow_{Exp} \quad \dfrac{\begin{matrix}D_2\\ \Box\Gamma \vdash \Box(A\ true)\end{matrix} \qquad \dfrac{\dfrac{}{\Gamma, a : A \vdash A\ true}\,Prem}{\Box\Gamma, a : A \vdash \Box(A\ true)}\,I\Box}{\Gamma \vdash A\ true}\,E\Box$$

By this expansion one shows how $E\Box$ provides all the information needed to reconstruct $\Box(A\ true)$. Computationally, it reconstructs the value on code A.[16] Soundness by local reduction on $\Diamond(A\ true)$:

$$\dfrac{\dfrac{\begin{matrix}D_1\\ \Gamma, x : A \vdash B\ true^*\end{matrix}}{\Box\Gamma, \Diamond(A\ true) \vdash \Diamond(B\ true)}\,I\Diamond \qquad \begin{matrix}E\\ \Gamma, \Delta \vdash A\ true^*\end{matrix}}{\Gamma, \Delta \vdash B\ true^*}\,E\Diamond \quad \Rightarrow_{Redex} \quad \begin{matrix}D_2\\ \Gamma, \Delta \vdash B\ true^*\end{matrix}$$

D_2 is justified from D_1 and E by the Hypothesis Rule and $I\Diamond$: by E, Γ, Δ in reduced form will contain at least one formula of $type_{inf}$, which justifies $true^*$ in D_2. Computationally, this reduction formalizes the naming of codes that are presented partially evaluated to program B. Finally, completeness by local expansion on $\Diamond(A\ true)$:

$$\underline{\begin{matrix}D_1\\ \Diamond\Gamma \vdash \Diamond(A\ true)\end{matrix}} \quad \Rightarrow_{Exp} \quad \dfrac{\begin{matrix}D_2\\ \Diamond\Gamma \vdash \Diamond(A\ true)\end{matrix} \qquad \dfrac{\dfrac{}{\Gamma, x : A \vdash A\ true}\,Hyp}{\Diamond\Gamma, \Diamond(A\ true) \vdash \Diamond(A\ true)}\,I\Diamond}{\Gamma \vdash A\ true^*}\,E\Diamond$$

[16] This formulation of the $\Box$-rules does not violate the meaning of hypotheses, as it is the case with the rules for necessity in [19]. On the other hand, given Definition 2, a side condition on multiple simultaneous substitutions is unavoidable, see [3].

This expansion shows how to reconstruct all the information needed to formulate $\Diamond(A\ true)$, as a partial evaluation of program A.

Model of this dependent types system is a weakening of the truth-values model.[17] Our truth-functional model considers its types as pairs $A = [a, \rightarrow]$, with a the verification term and $\rightarrow$ the evaluation function:

- $A = [a, \rightarrow] = \{1\}$ if $x \rightarrow a = 1$ and $A : type = 1$
- $A = [a, \rightarrow] = \emptyset$ if $x \rightarrow a = undefined$ and $A : type_{inf} = 1$
- $A = [a, \rightarrow] = \{0\}$ if $x \rightarrow a = 0$ and $A : type = 0$

The models for $type_{inf}$ admits undefinability, hence it preserves only symmetricity on the standard models and the partition is no longer satisfied, i.e. inhabitness is not guaranteed.[18]

4 Conclusions

Our modal type system allows refutable truths in a constructive setting. The main application is the modeling of knowledge processes with embedded communication processes intended as refutable contents in a distributed or staged format: this is especially interesting in view of a multi-modal version of this language, where the dependency relation can be interpreted as communication among nodes of a trusted network. The comparison with staged and distributed processing is completed by indexing of local processes and the interpretation of modalities as code mobility. The extension to a multi-conclusion inference relation is the next obvious step for this modal type-theory.

References

1. N. Alechina, M. Mendler, V. de Paiva, and E. Ritter. Categorical and Kripke Semantics for Constructive S4 Modal Logic. In *Proceedings of the 15th International Workshop on Computer Science Logic*, volume 2142 of *Lecture Notes In Computer Science*, pages 292 – 307, 2001.
2. G. Bellin, V. de Paiva, and E. Ritter. Extended Curry-Howard Correspondence for a Basic Constructive Modal Logic. preprint; presented at M4M-2, ILLC, UvAmsterdam, 2001, 2001.
3. G.M. Bierman and V. de Paiva. Intuitionistic necessity revisited. Technical Report CSRP-96-10, School of Computer Science, University of Birmingham, 1996.
4. R. Davies and F. Pfenning. A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604, 2001.
5. V. de Paiva. *Natural Deduction and Context as (Constructive) Modality*, P. Balckburn et al. (eds.), CONTEXT 2003, volume 2680 of *Lecture Notes in Artificial Intelligence*, pp. 116-129. Springer Verlag, 2003.

[17] The latter is given by the category of contexts as the poset $\{1, 0\}$ that satisfies inhabitness by at most one element and intensional identity types.

[18] A weakening of the PER models, that could be called 'super-modest types'.

6. M. Fairtlough and M. Mendler. On the logical content of computational type theory: A solution to Curry's problem. In P. Callaghan, Z. Luo, and J. McKinna, editors, *Types for Proofs and Programs*, volume 2277 of *Lectures Notes in Computer Science*, pages 63–78. Springer Verlag, 2002.
7. M. Fitting. *Basic modal logic*, volume 4 of *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 368–449. Oxford University Press, 1994.
8. R. Hakli and S. Negri. Does the deduction theorem fail for modal logic? Manuscript, to appear, 2008.
9. N.D. Jones, C. Gomard, and P. Sestoft. *Partial Evaluation and Automatic Program Generation*. International Series in Computer Science. Prentice-Hall International, 1993.
10. A. Kolmogorov. On the principle of excluded middle. In J. Van Heijenoort, editor, *From Frege to Gödel: a source book in mathematical logic 1879-1931*, pages 414–437. Harvard University Press, 1967.
11. P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.
12. P. Martin-Löf. On the meaning of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.
13. M. Mendler and V. de Paiva. Constructive CK for Contexts. In *Proceedings of the first Workshop on Context Representation and Reasoning - CONTEXT05*, 2005.
14. J. Moody. Modal logic as a basis for distributed computation. Technical Report CMU-CS-03-194, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, USA, 2003.
15. T. Murphy, K. Crary, R. Harper, and F. Pfenning. A symmetric modal lambda calculus for distributed computing. In H. Ganzinger, editor, *Proceedings of the 19th Annual Symposium on Logic in Computer Science (LICS'04)*, pages 286–295. IEEE Computer Society Press, 2004.
16. A. Nanevski, F. Pfenning, and B. Pientka. Contextual modal type theory. *ACM Transactions on Computational Logic*, 9(3):1–48, 2008.
17. F. Pfenning. Intensionality, extensionality, and proof irrelevance in modal type theory. In J.Y. Halpern, editor, *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science*, pages 221–230. IEEE Computer Society Press, 2001.
18. F. Pfenning and R. Davies. A judgemental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001.
19. D. Prawitz. *Natural Deduction*. Almqvist & Wiksell, 1965.
20. G. Primiero. A constructive modal semantics for contextual verification. In A. Mileo and J. Delgrande, editors, *Proceedings of the First international Workshop on Logic-based Interpretation of Context: Modelling and Applications*, volume 550 of *CEUR Workshop Proceedings*, pages 33–35, 2009. Full Paper available as Technical Report, Centre for Logic and Philosophy of Science, Ghent University.
21. G. Primiero. Epistemic modalities. In G. Primiero and S. Rahman, editors, *Acts of Knowledge: History, Philosophy and Logic*, volume 9 of *Tributes*, pages 207–232. College Publications, 2009.
22. A. Tasistro. Formulation of Martin-Löf's Type Theory with Explicit Substitutions. Lic thesis, Department of Computing Science, Chalmers University of Technology and University of Göteborg, 1993.
23. A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction*, volume I,II. North-Holland, Amsterdam, 1988.
24. D. Wijesekera. Constructive modal logics I. *Annals of Pure and Applied Logic*, 50:271–301, 1990.